Line Follower MRT 3 / HUNA

Research & Development TEAM

Menbuat Program Line Follower

(MRT 3 / HUNA)

Dasar dari pemrograman maze.h pada maze solving MRT yaitu pemrograman robot line follower, karena maze solving merupakan line follower yang dimodifikasi sedemikian rupa sehingga menghasilkan tantangan tantangan baru dalam pemrogramannya.

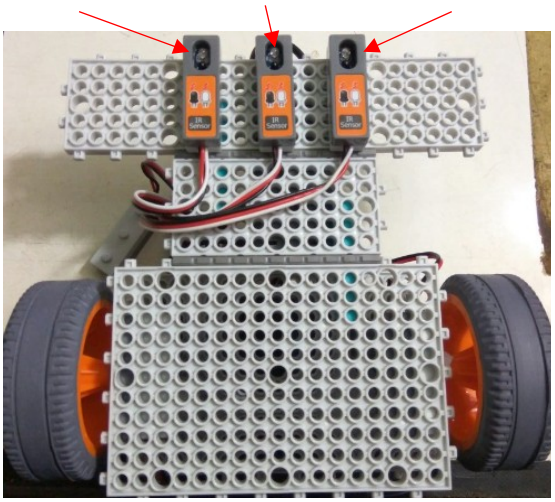
Oleh karena itu sebelum kita membahas lebih jauh tentang maze solving, sebaiknhya kita pelajari dahulu apa itu robot line follower? bagaimana cara kerja robot line follower? dan bagaimana cara memprogram line follower pada MRT.

A. Robot Line Follower

Robot line follower adalah robot yang dapat bergerak sesuai garis yang ditentukan, baik garis hitam maupun garis putih. Biasanya robot line follower mengunakan sensor infrared (IR) untuk mendeteksi warna dari jalur yang akan dilaluinya.
Pada beberapa kasus sensor IR yang digunakan berjumlah satu atau dua sensor saja, namun biasanya agar pergerakan robot lebih halus maka digunakan tiga sensor IR. Pada pembahasan kali ini kita akan menggunakan tiga sensor IR (sensor kiri, sensor tengah dan sensor kanan).

B. Konfigurasi line follower pada MRT

IR2 *(IN2)* **IR3** *(IN3)* **IR4** *(IN4)*
Sensor kanan **Sensor Tengah** **Sensor kiri**



Tampak Bawah

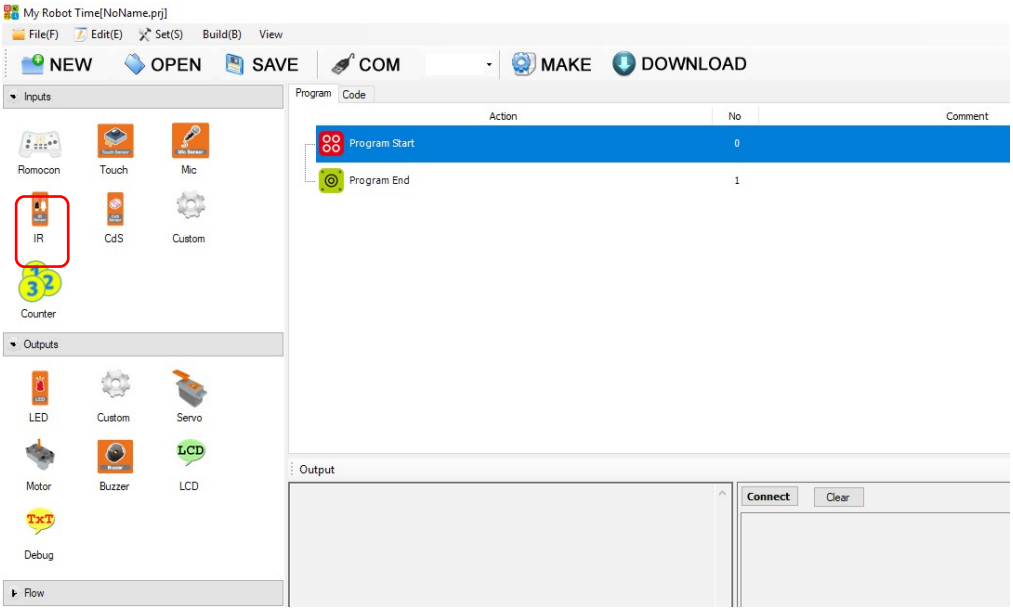
- IR 2 (sebelah kiri IR 1) dihubungkan ke Pin Input 2
- IR 3 (IR yang tengah) dihubunkan ke Pin Input 3
- IR 4 (sebelah kiri IR 3) dihubungkan ke Pin Input 4

C. Algoritma (Alur Program) line follower MRT

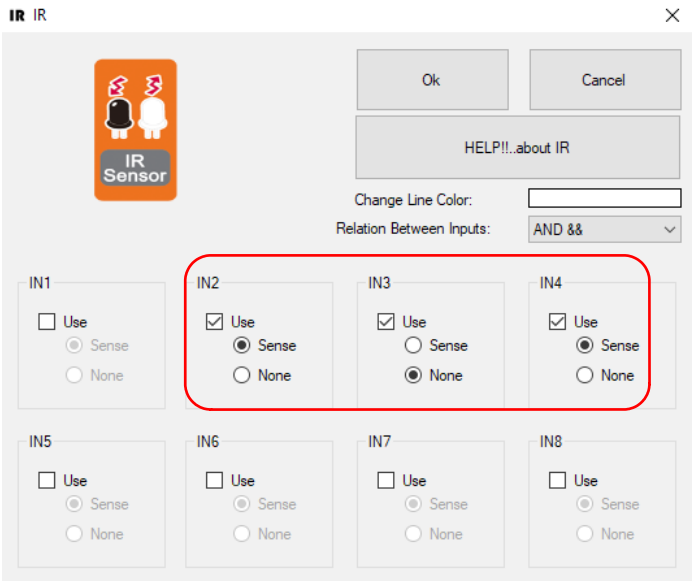
1) Inisialisasi program
2) Jika sensor tengah mendeteksi garis hitam, maka robot bergerak maju
3) Jika sensor kiri mendeteksi garis hitam, maka robot berbelok ke kiri
4) Jika sensor kanan mendeteksi garis hitam, maka robot berbelok ke kanan
5) Ulangi lagi ke langkah no 2.

D. Pemrograman line follower pada MRT

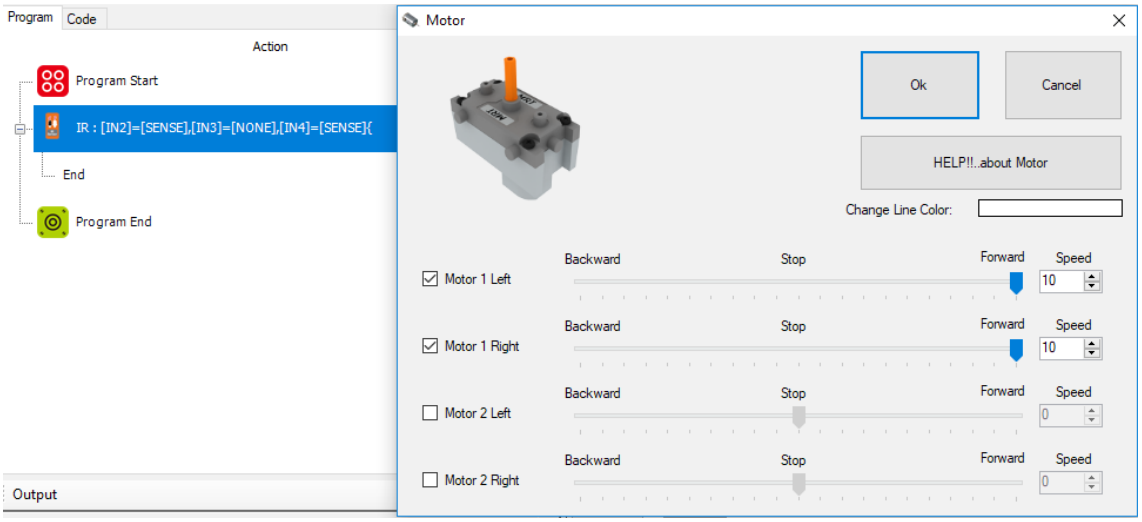
1) Buka software MRT => Klik "Program Start" => klik gambar IR pada blok input.



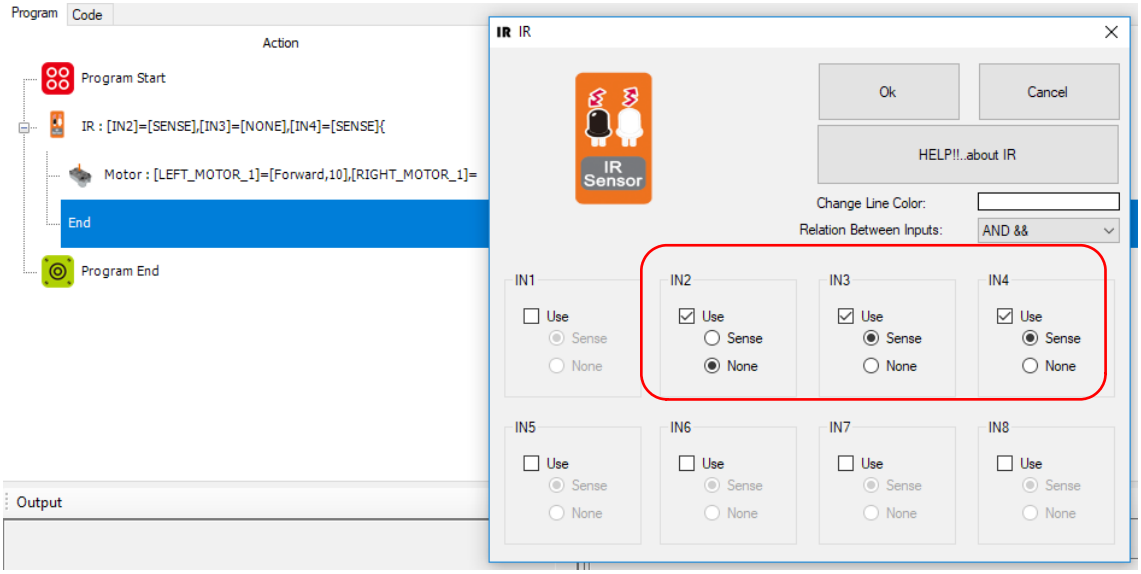
2) Aktifkan input dua, tiga dan empat, dan pilih mode sense pada input 2, mode none pada imput 3 dan mode sense pada input 4 (Ket: sensor tengah mendeteksi garis hitam).



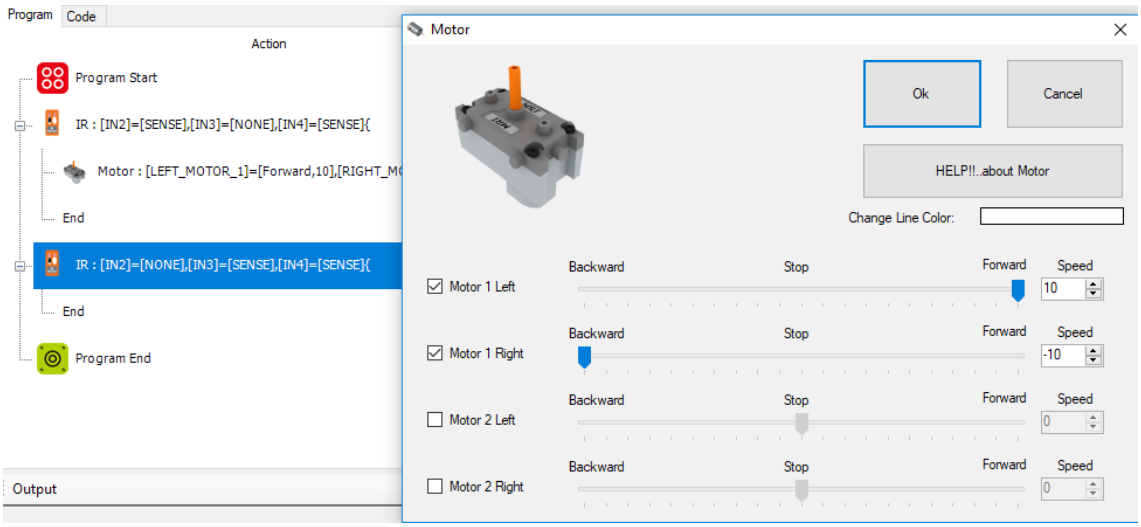
3) Aktifkan "motor 1 left" isikan nilai speed nya antara 1 s/d 10 (disarankan 8). Dan aktifkan juga "motor 1 right" isikan nilai speednya sama dengan speed "motor 1 left".



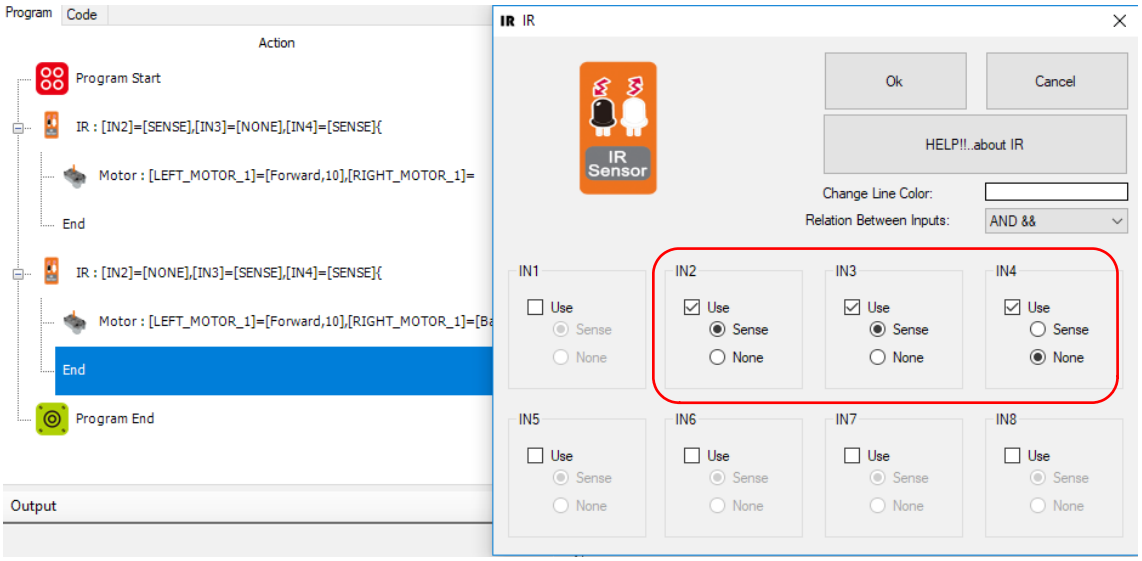
4) Klik "End" pada baris program, lalu aktifkan input dua, tiga dan empat, dan pilih mode none pada input 2, mode sense pada imput 3 dan mode sense pada input 4 (Ket: sensor kanan mendeteksi garis hitam).



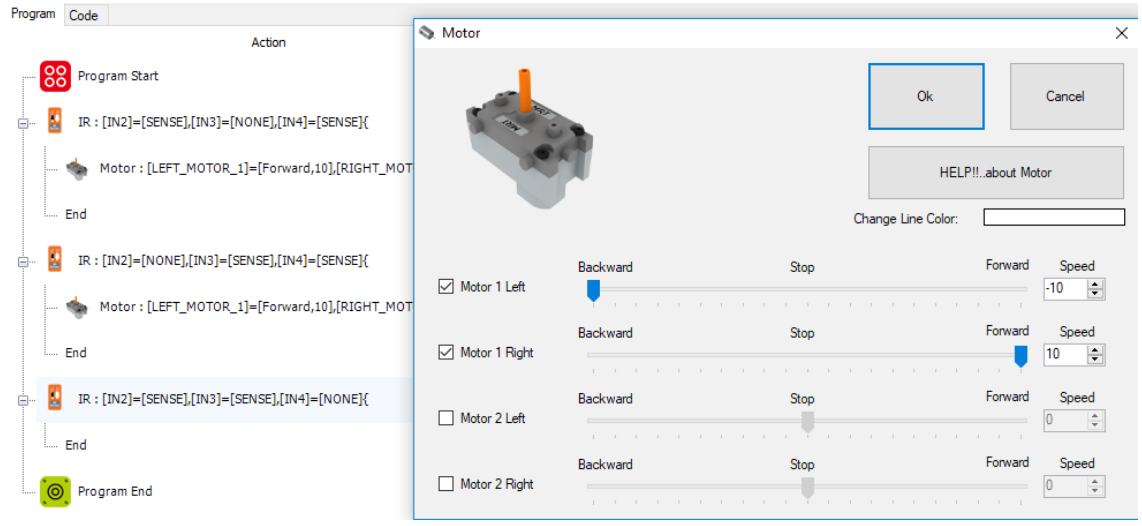
5) Aktifkan “motor 1 left” isikan nilai speed nya antara 1 s/d 10 (disarankan 8). Dan aktifkan juga “motor 1 right” isikan nilai speednya antara -1 s/d -10 (disarankan -8).



6) Klik “End” pada baris program, lalu aktifkan input dua, tiga dan empat, dan pilih mode sense pada input 2, mode sense pada imput 3 dan mode none pada input 4 (Ket: sensor kiri mendeteksi garis hitam).



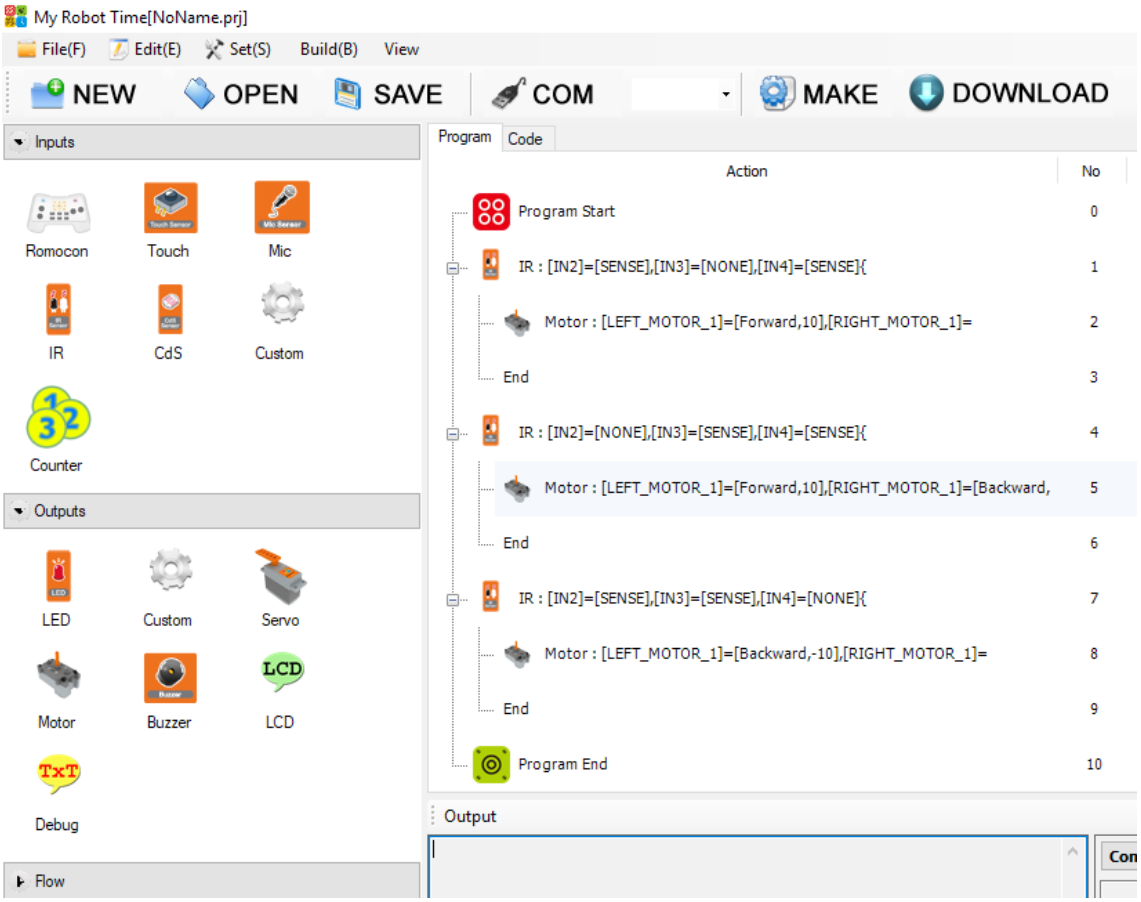
7) Aktifkan “motor 1 left” isikan nilai speed nya antara -1 s/d -10 (disarankan -8). Dan aktifkan juga “motor 1 right” isikan nilai speednya antara 1 s/d 10 (disarankan 8).



8) Program siap di download ke board MRT 3.

Line Follower MRT 3 / HUNA

Research & Development TEAM

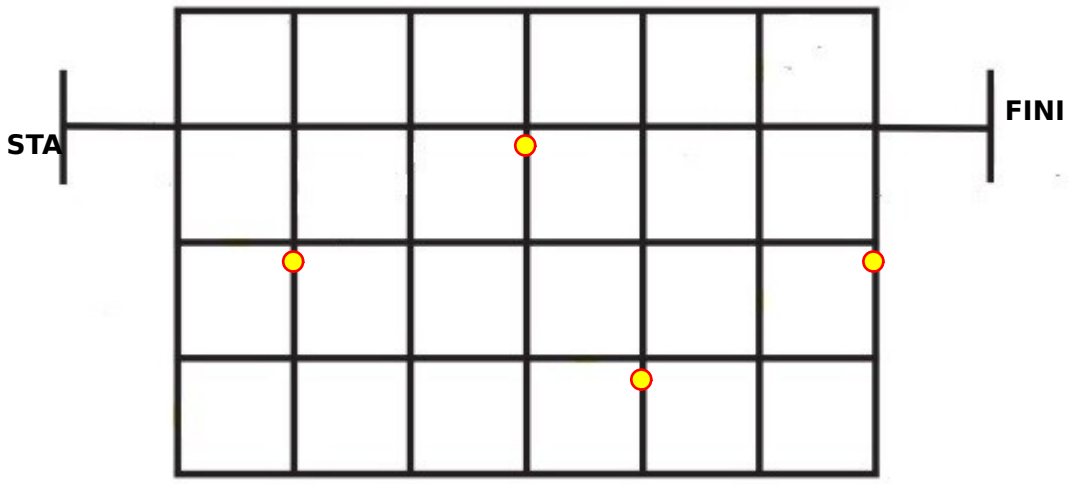


Maze Solving MRT 3 / HUNA

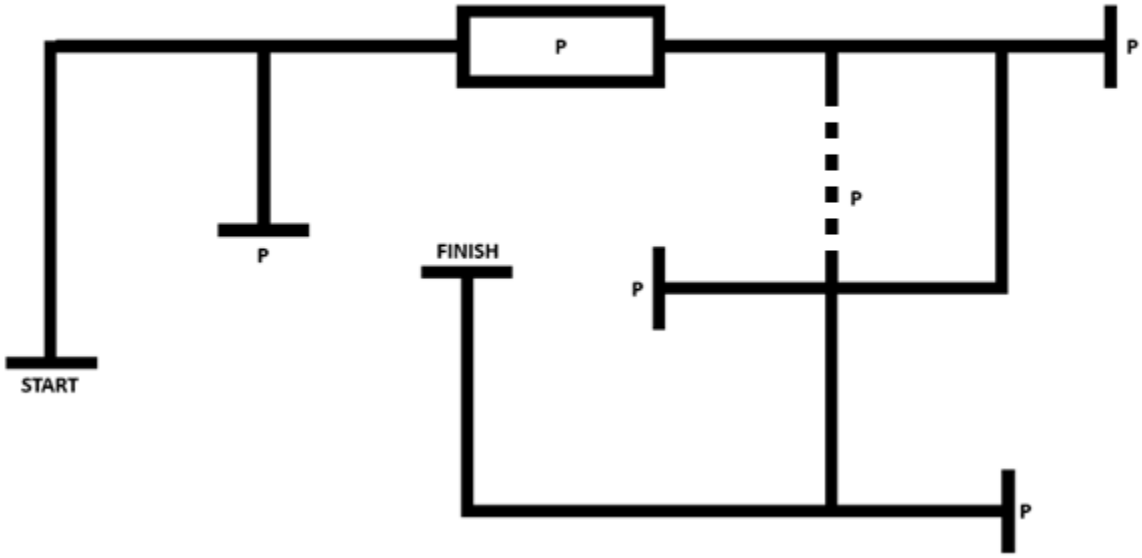
Research & Development TEAM

Cara Menggunakan “maze.h” Pada Lintasan Maze Solving

A. Contoh jenis arena maze solving

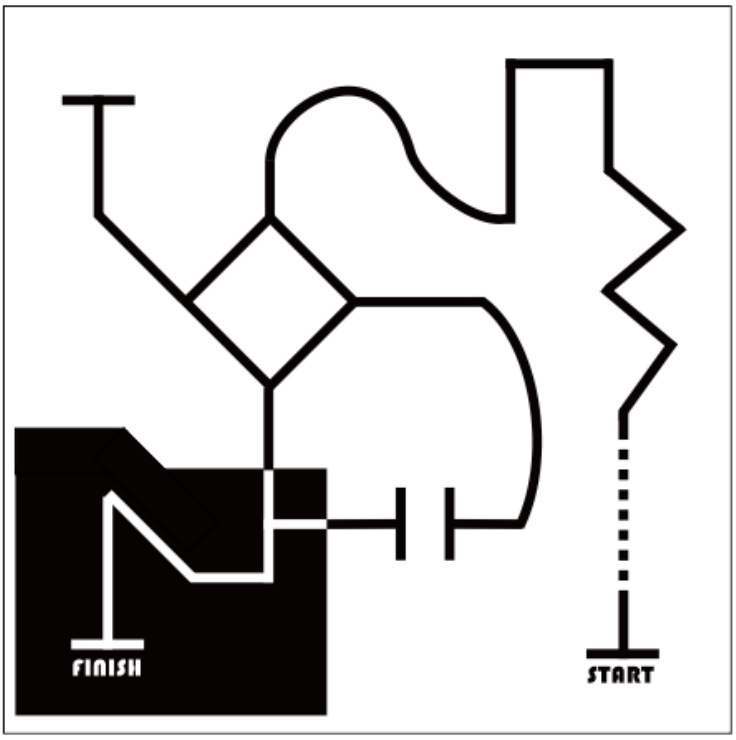


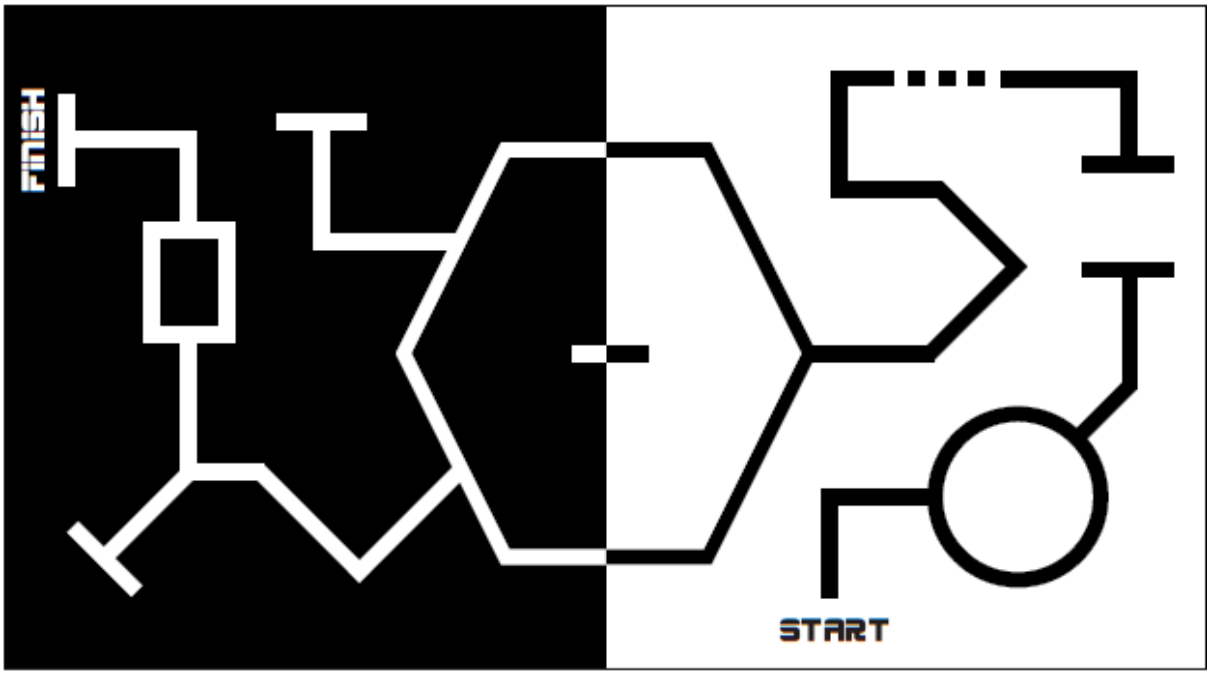
Pada jenis maze solving diatas peserta harus memprogram robot agar menabrak pipa kuning (ditentukan oleh panitia) dan mengakhiri lajunya di garis finish. Pemenang ditentukan oleh point terbesar, jika point sama maka ditentukan dengan waktu tercepat.



Pada jenis maze solving diatas peserta harus memprogram robot agar mendapatkan point sebanyak-banyaknya (6 titik point) dan mengakhiri jalannya sampai garis finish. Pemenang ditentukan oleh

point terbesar, jika point sama maka ditentukan dengan waktu tercepat.



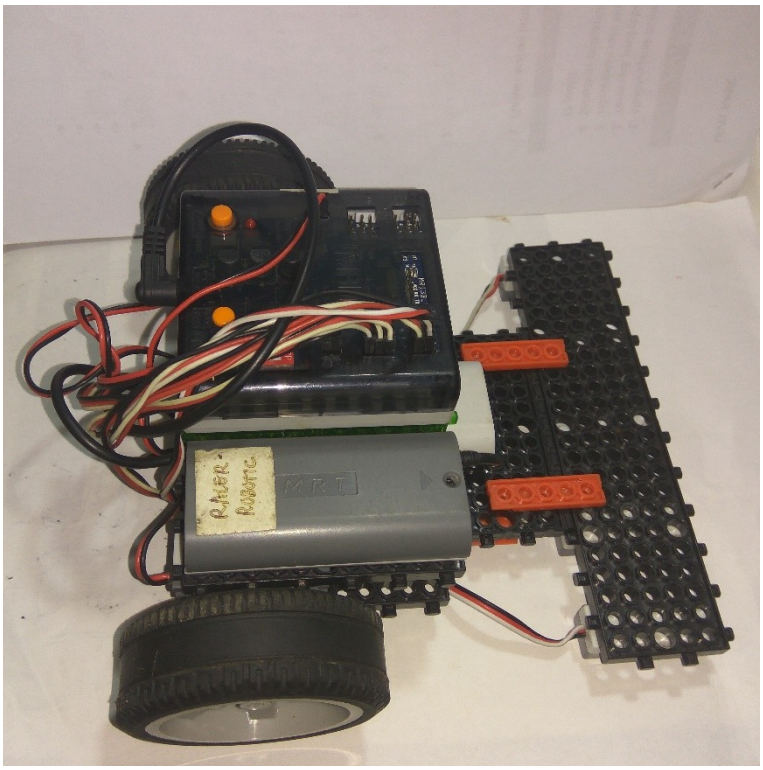


Pada jenis maze solving diatas peserta harus memprogram robot agar mendapatkan point sebanyak-banyaknya (6 titik point) dan mengakhiri jalannya sampai garis finish. Pemenang ditentukan oleh

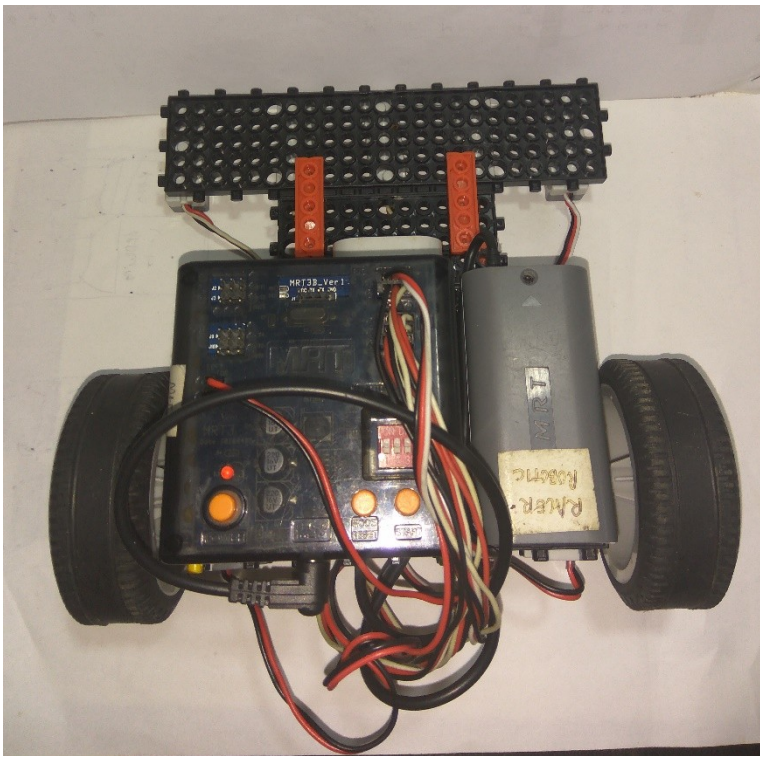
point terbesar, jika point sama maka ditentukan dengan waktu tercepat.

B. Desain robot maze solving

Tampak Samping



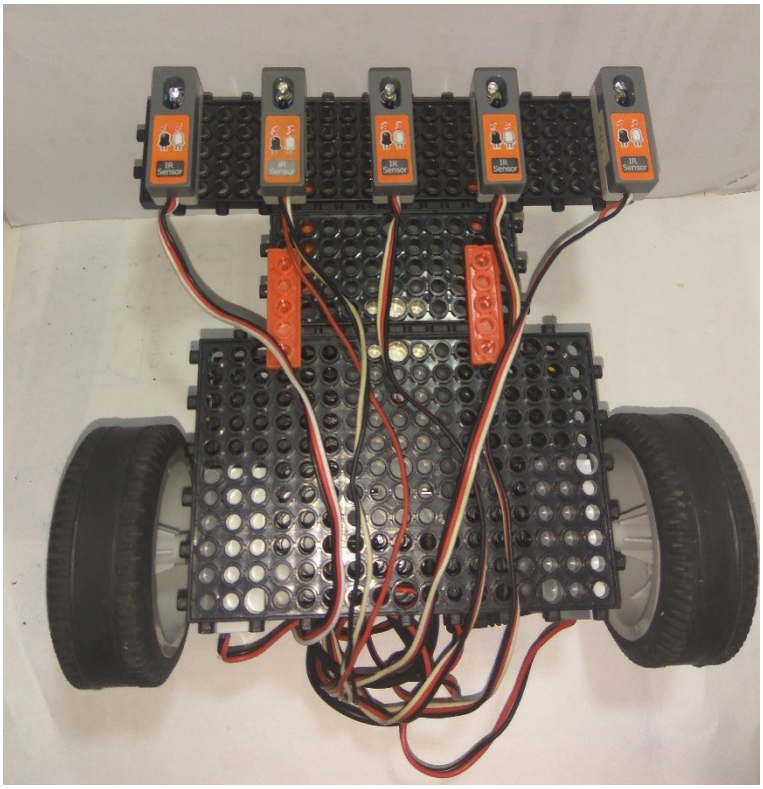
Tampak Atas



Maze Solving MRT 3 / HUNA

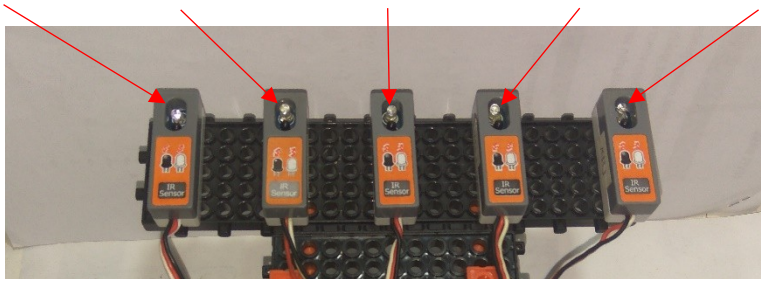
Research & Development TEAM

Tampak Bawah



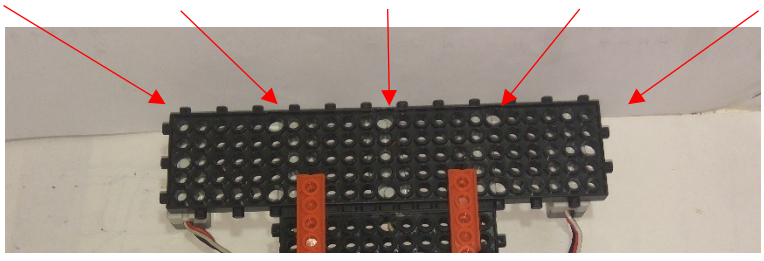
Posisi IR (Tampak Bawah)

IR1 *(IN1)* **IR2** *(IN2)* **IR3** *(IN3)* **IR4** *(IN4)* **IR5** *(IN5)*



Posisi IR (Tampak Atas)

IR5 *(IN5)* **IR4** *(IN4)* **IR3** *(IN3)* **IR2** *(IN2)* **IR1** *(IN1)*



Maze Solving MRT 3 / HUNA

Research & Development TEAM

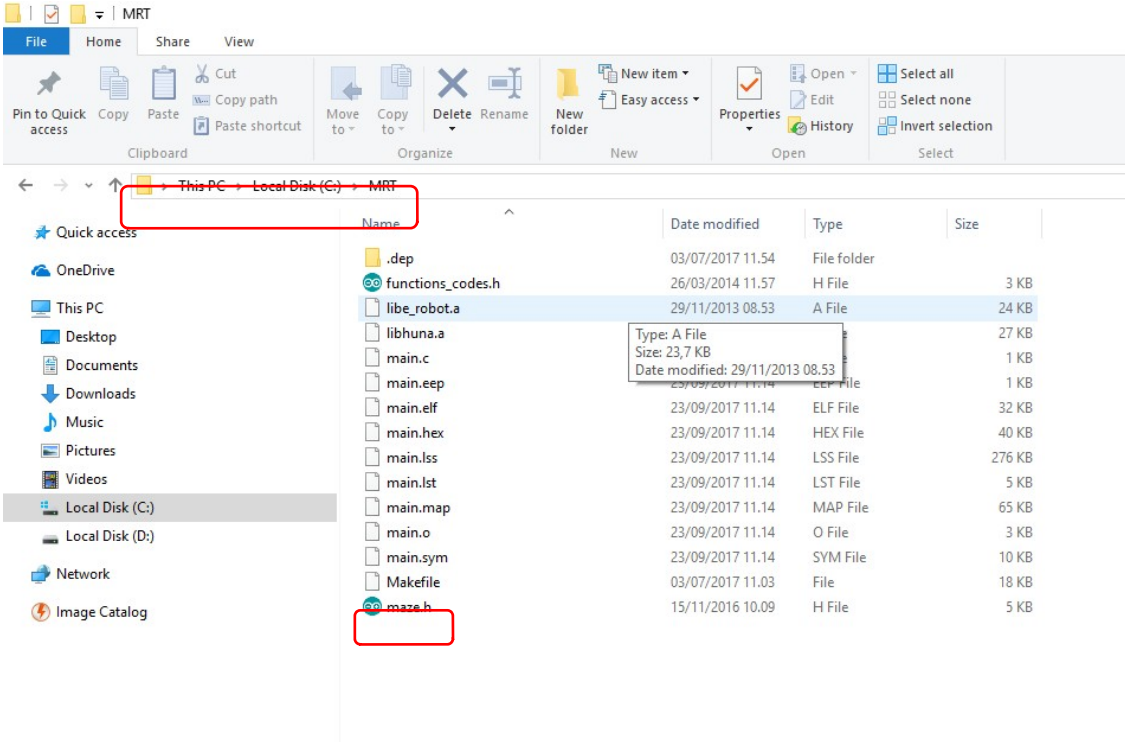
Pada pemrograman maze solving menggunakan maze.h, sensor IR yang digunakan sebanyak lima buah. Dengan mapped pin sebagai berikut (tampak atas):

1. IR 1 (IR paling kanan) dihubungkan ke Pin Input 1
2. IR 2 (sebelah kiri IR 1) dihubungkan ke Pin Input 2
3. IR 3 (IR yang tengah) dihubunkan ke Pin Input 3
4. IR 4 (sebelah kiri IR 3) dihubungkan ke Pin Input 4
5. IR 5 (IR paling kiri) dihubungkan ke Pin Input 5

C. Langkah-langkah sebelum memprogram

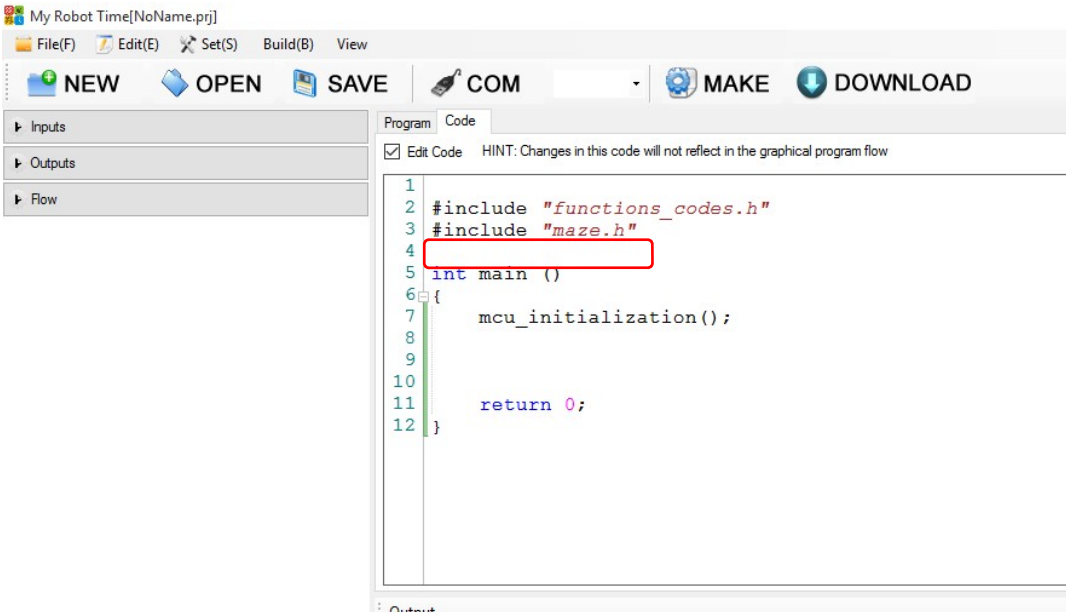
Sebelum memprogram menggunakan maze.h ada beberapa langkah yang harus dilakukan terlebih dahulu, berikut :

1. Simpan file maze.h ke directory: C:\MRT

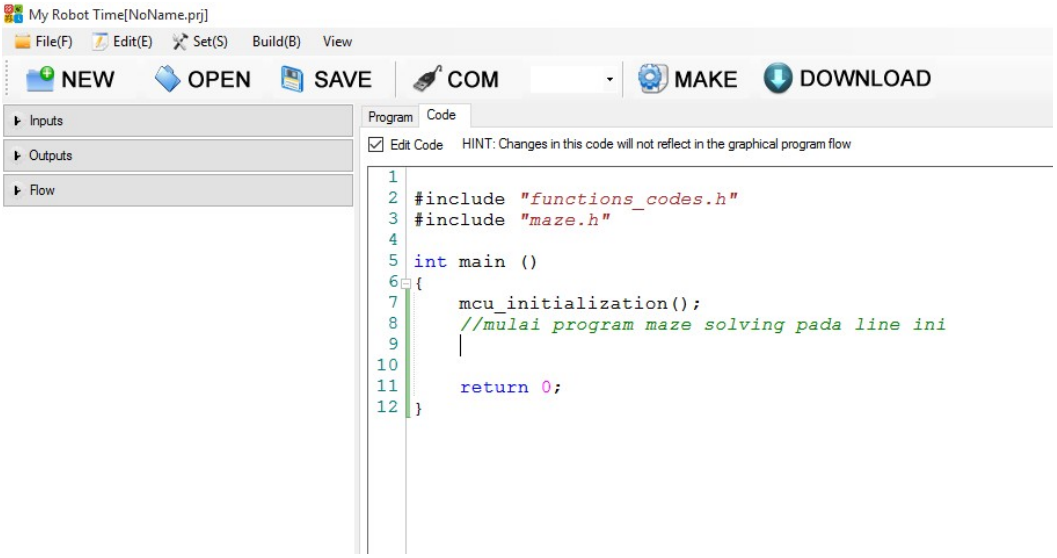


2. Buka aplikasi MRT3 => Pilih pemrograman dengan mode "Code" => klik checkbox "Edit Code"

3. Ketik *#include "maze.h"* diatas sintax *int main ()* => hapus sintax while (1) dan dua sintax dibawahnya.



4. Program maze solving menggunakan maze.h siap dimulai



D. Dasar pemrograman maze.h

Inti dari pemrograman maze.h yaitu, robot mengikuti garis sampai menemui tikungan atau persimpangan didepannya.

a) Tikungan kanan hitam

Tikungan kanan hitam terjadi jika sensor IR 1, IR2 dan IR3 mendeteksi garis hitam. Sintax untuk menggerakan robot sampai menemui tikungan kanan hitam yaitu,

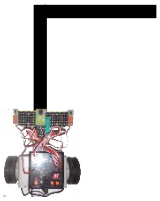
ikutgarishitam(12,_,_);

Speed motor kiri

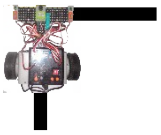
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    ikutgarishitam(12,8,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



b) Tikungan kanan putih

Tikungan kanan putih terjadi jika sensor IR 1, IR2 dan IR3 mendeteksi garis putih. Sintax untuk menggerakan robot sampai menemui tikungan kanan putih yaitu,

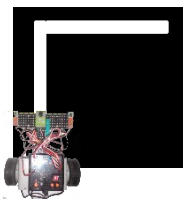
ikutgarisputih(12,_,_);

Speed motor kiri

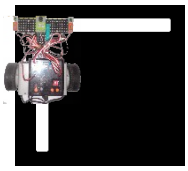
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    ikutgarisputih(12,8,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



c) Tikungan kiri hitam

Tikungan kiri hitam terjadi jika sensor IR 4, IR5 dan IR3 mendeteksi garis hitam. Sintax untuk menggerakan robot sampai menemui tikungan kiri hitam yaitu,

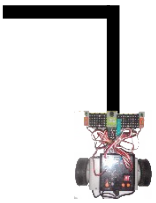
ikutgarishitam(45,_,_);

Speed motor kiri

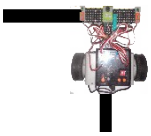
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    ikutgarishitam(45,8,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



d) Tikungan kiri putih

Tikungan kiri putih terjadi jika sensor IR 4, IR5 dan IR3 mendeteksi garis putih. Sintax untuk menggerakan robot sampai menemui tikungan kiri putih yaitu,

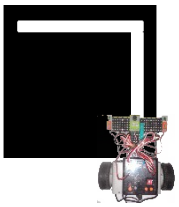
ikutgarisputih(45,_,_);

Speed motor kiri

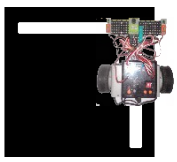
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    ikutgarisputih(45,8,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



e) Tikungan kanan tajam hitam

Tikungan kanan tajam hitam terjadi jika sensor IR 2 dan IR 3 mendeteksi garis hitam. Sintax untuk menggerakan robot sampai menemui tikungan kanan tajam hitam yaitu,

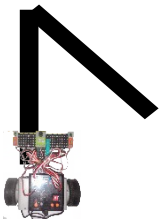
ikutgarishitam(23,_,_);

Speed motor kiri

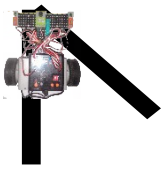
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    ikutgarishitam(23,8,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



f) Tikungan kanan tajam putih

Tikungan kanan tajam putih terjadi jika sensor IR 2 dan IR 3 mendeteksi garis putih. Sintax untuk menggerakan robot sampai menemui tikungan kanan tajam putih yaitu,

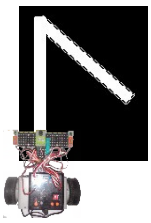
ikutgarisputih(23,_,_);

Speed motor kiri

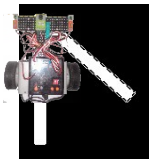
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    ikutgarisputih(23,8,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



g) Tikungan kiri tajam hitam

Tikungan kiri tajam hitam terjadi jika sensor IR 3 dan IR 4 mendeteksi garis hitam. Sintax untuk menggerakan robot sampai menemui tikungan kiri tajam hitam yaitu,

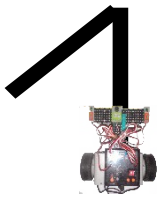
ikutgarishitam(34,_,_);

Speed motor kiri

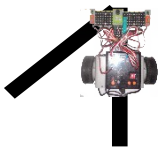
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    ikutgarishitam(34,8,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



h) Tikungan kiri tajam putih

Tikungan kiri tajam putih terjadi jika sensor IR 3 dan IR 4 mendeteksi garis putih. Sintax untuk menggerakan robot sampai menemui tikungan kiri tajam putih yaitu,

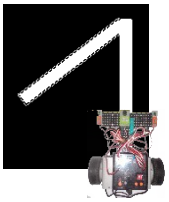
ikutgarisputih(34,_,_);

Speed motor kiri

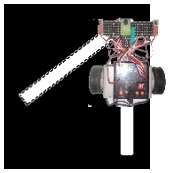
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    ikutgarisputih(34,8,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



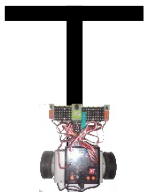
i) Persimpangan hitam

Persimpangan hitam terjadi jika sensor IR 2 dan IR4 mendeteksi garis hitam.
Sintax untuk menggerakan robot sampai menemui simpangan hitam yaitu,
ikutgarishitam(24,_,_);

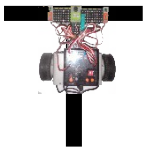
Speed motor kiri
Speed motor kanan

```
1
2  #include "functions_codes.h"
3  #include "maze.h"
4
5  int main ()
6  {
7      mcu_initialization();
8      ikutgarishitam(24,8,8);
9      return 0;
10 }
```

Kondisi awal



Kondisi akhir



j) Persimpangan putih

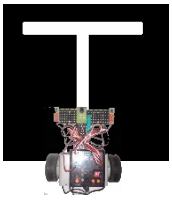
Persimpangan putih terjadi jika sensor IR 2 dan IR4 mendeteksi garis putih.
Sintax untuk menggerakan robot sampai menemui simpangan putih yaitu,
ikutgarisputih(24,_,_);

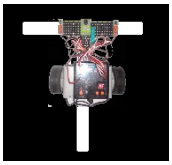
Speed motor kiri
Speed motor kanan

```
1
2  #include "functions_codes.h"
3  #include "maze.h"
4
5  int main ()
6  {
7      mcu_initialization();
8      ikutgarisputih(24,8,8);
9      return 0;
10 }
```

Kondisi awal

Kondisi akhir





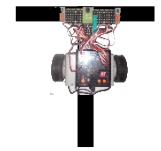
k) Belok kanan

Belok kanan digunakan untuk memutarkan robot agar menghadap ke bagian kanan. Dalam sintax *"kanan(_,_,_);"* terdapat nilai untuk delay, nilai ini berpengaruh terhadap seberapa tajam robot berbelok. Semakin besar nilai delay maka semakin tajam robot berbelok.
Sintax untuk menghadapkan robot ke arah kanan yaitu,
kanan(_,_,_);

Speed motor kiri Delay
Speed motor kanan

```
1
2  #include "functions_codes.h"
3  #include "maze.h"
4
5  int main ()
6  {
7      mcu_initialization();
8      kanan(8,5,250);
9      return 0;
10 }
```

Kondisi awal Kondisi akhir

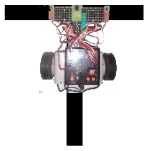


l) Belok kiri
Belok kiri digunakan untuk memutarkan robot agar menghadap ke bagian kiri. Dalam sintax *"kiri(_,_,_);"* terdapat nilai untuk delay, nilai ini berpengaruh terhadap seberapa tajam robot berbelok. Semakin besar nilai delay maka semakin tajam robot berbelok.
Sintax untuk belok kiri yaitu,
kiri(_,_,_);

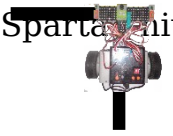
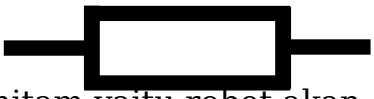
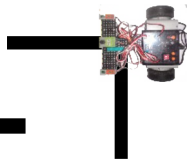
```
2  #include "functions_codes.h"
3  #include "maze.h"
4
5  int main ()
6  {
7      mcu_initialization();
8      kiri(5,8,250);
9      return 0;
10 }
```

Speed motor kiri Delay
Speed motor kanan

Kondisi awal Kondisi akhir



m) Spartan hitam

Cara sintax spartan hitam yaitu robot akan:

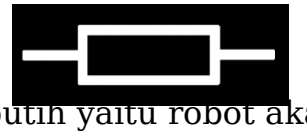
- maju ketika sensor IR2 dan IR4 mendeteksi garis putih, sampai IR2 dan IR4 mendeteksi garis hitam
- jika IR2 mendeteksi hitam robot akan belok ke kiri sampai IR2 dan IR4 mendeteksi garis putih (meluruskan posisi)
- jika IR4 mendeteksi hitam robot akan belok ke kanan sampai IR2 dan IR4 mendeteksi garis putih (meluruskan posisi)

Sintax untuk melewati spartan hitam yaitu;
spartanhitam();

```
1
2  #include "functions_codes.h"
3  #include "maze.h"
4
5  int main ()
6  {
7      mcu_initialization();
8      spartanhitam();
9      return 0;
10 }
```

Kondisi awal Kondisi akhir

n) Spartan putih



Cara sintax spartan putih yaitu robot akan:

- maju ketika sensor IR2 dan IR4 mendeteksi garis hitam, sampai IR2 dan IR4 mendeteksi garis putih

- jika IR2 mendeteksi putih robot akan belok ke kiri sampai IR2 dan IR4 mendeteksi garis hitam (meluruskan posisi)
- jika IR4 mendeteksi putih robot akan belok ke kanan sampai IR2 dan IR4 mendeteksi garis hitam (meluruskan posisi)

Sintax untuk melewati spartan putih yaitu,
spartanputih();

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    spartanputih();
    return 0;
}
```

Kondisi awal

itam

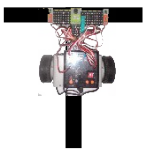
Kondisi akhir

Fungsi dari balik kanan hitam adalah membelokan robot ke kanan untuk memutar arah dari posisi semula. Inti dari sintax *"balikkananhitam(_,_);"* yaitu robot balik arah sampai sensor IR3 mendeteksi garis hitam.
Sintax untuk balik kanan hitam yaitu,
balikkananhitam(_,_);

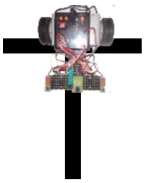
Speed motor kiri
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    balikkananhitam(8,-10);
    return 0;
}
```

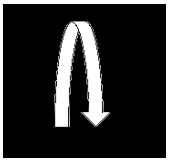
Kondisi awal



Kondisi akhir



p) Balik kanan putih



Fungsi dari balik kanan putih adalah membelokan robot ke kanan untuk memutar arah dari posisi semula. Inti dari sintax *"balikkananputih(_,_);"* yaitu robot balik arah sampai sensor IR3 mendeteksi garis putih.

Sintax untuk balik kanan putih yaitu,

balikkananputih(_,_);

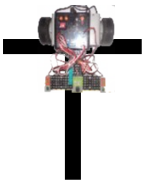
Speed motor kiri

Speed motor kanan

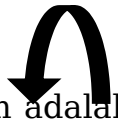
```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    balikkananputih(8,-10);
    return 0;
}
```

Kondisi awal

Kondisi akhir



q) Balik kiri hitam



Fungsi dari balik kiri hitam adalah membelokan robot ke kiri untuk memutar arah dari posisi semula. Inti dari sintax *"balikkirihitam(_,_);"* yaitu robot balik arah sampai sensor IR3 mendeteksi garis hitam.

Sintax untuk balik kiri hitam yaitu,

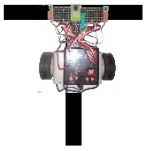
balikkirihitam(_,_);

Speed motor kiri

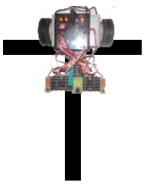
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    balikkirihitam(-10,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



r) Balik kiri putih

Fungsi dari balik kiri putih adalah membelokan robot ke kiri untuk memutar arah dari posisi semula. Inti dari sintax *"balikkiriputih(_,_);"* yaitu robot balik arah sampai sensor IR3 mendeteksi garis putih.

Sintax untuk balik kiri hitam yaitu,

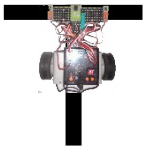
balikkiriputih(_,_);

Speed motor kiri

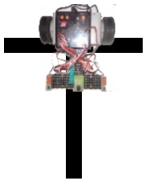
Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    balikkiriputih(-10,8);
    return 0;
}
```

Kondisi awal



Kondisi akhir



s) Jump hitam

Fungsi dari jump hitam yaitu, robot dapar melaju ke simpangan berikutnya tanpa harus mengikuti garis hitam. Inti dari sintax *"jumphitam(_,_);"* yaitu robot maju sampai sensor IR2 dan IR4 mendeteksi garis hitam.

Sintax untuk jump hitam yaitu,

jumphitam(_,_);

Speed motor kiri

Speed motor kanan

```
#include "functions_codes.h"
#include "maze.h"

int main ()
{
    mcu_initialization();
    jumphitam(8,8);
    return 0;
}
```

Kondisi awal

Kondisi akhir

t) Jump putih

Fungsi dari jump putih yaitu, robot dapar melaju ke simpangan berikutnya tanpa harus mengikuti garis putih. Inti dari sintax *"jumpputih(_,_);"* yaitu robot maju sampai sensor IR2 dan IR4 mendeteksi garis putih.

Sintax untuk jump putih yaitu,

jumpputih(_,_);

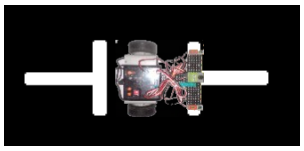
Speed motor kiri

Speed motor kanan

```
1
2  #include "functions_codes.h"
3  #include "maze.h"
4
5  int main ()
6  {
7      mcu_initialization();
8      jumpputih(8,8);
9      return 0;
10 }
```

Kondisi awal

Kondisi akhir



u) Stop

Stop digunakan untuk menghentikan robot dalam kondisi tertentu atau pada akhir program dengan nilai waktu tertentu. Satuan nilai stop pada sintax *"stop(_);"* yaitu milli detik (ms).

Sintax untuk stop yaitu,

stop(_);

Delay

```
1
2  #include "functions_codes.h"
3  #include "maze.h"
4
5  int main ()
6  {
7      mcu_initialization();
8      ikutgarishitam(24,8,8);
9      stop(5000);
10     balikkirihitam(-10,8);
11     stop(3000);
12     return 0;
13 }
```

Setelah robot ikut garis hitam 24, robot akan

Setelah robot balik kiri hitam robot akan berhenti seterusnya,

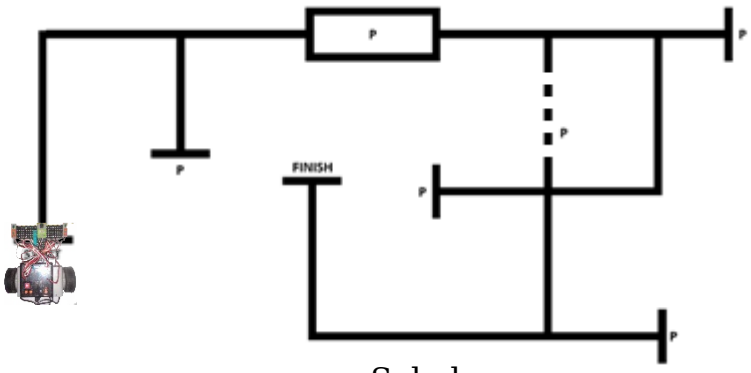
berhenti selama 5 detik lalu robot akan balik kiri hitam.

karena sintax stop berada diakhir program.

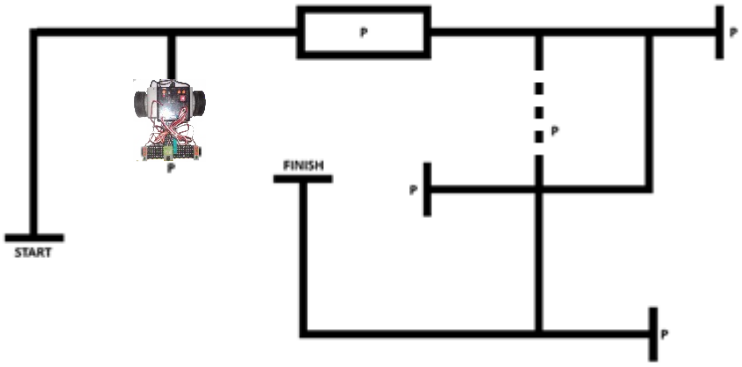
E. Contoh penyelesaian program maze.h

1) Mengambil point ke 1

```
1
2  #include "functions_codes.h"
3  #include "maze.h"
4
5  int main ()
6  {
7      mcu_initialization();
8      ikutgarishitam(12,8,8);
9      kanan(8,5,250);
10     ikutgarishitam(12,8,8);
11     kanan(8,5,250);
12     ikutgarishitam(24,8,8);
13     stop(3000);
14     return 0;
15 }
```



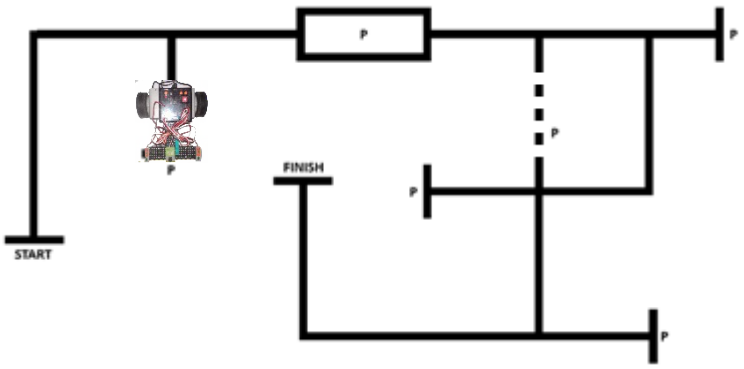
Sebelum program

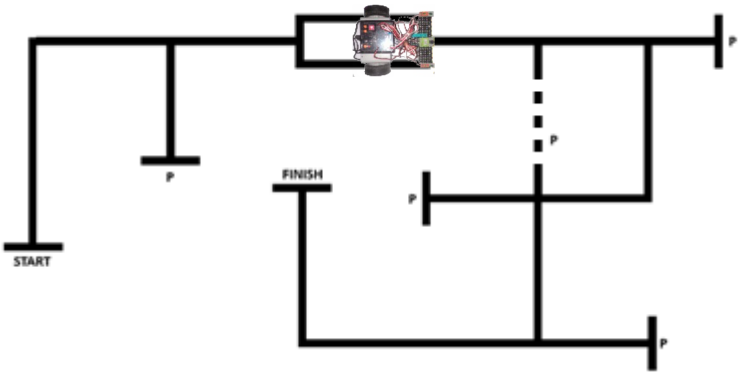


Ssetelah di program

2) Mengambil point ke 2 dari posisi point ke 1

```
1
2  #include "functions_codes.h"
3  #include "maze.h"
4
5  int main ()
6  {
7      balikkananhitam(8,-10);
8      ikutgarishitam(24,8,8);
9      kanan(8,5,250);
10     ikutgarishitam(24,8,8);
11     spartanhitam();
12     stop(3000);
13     return 0;
14 }
```

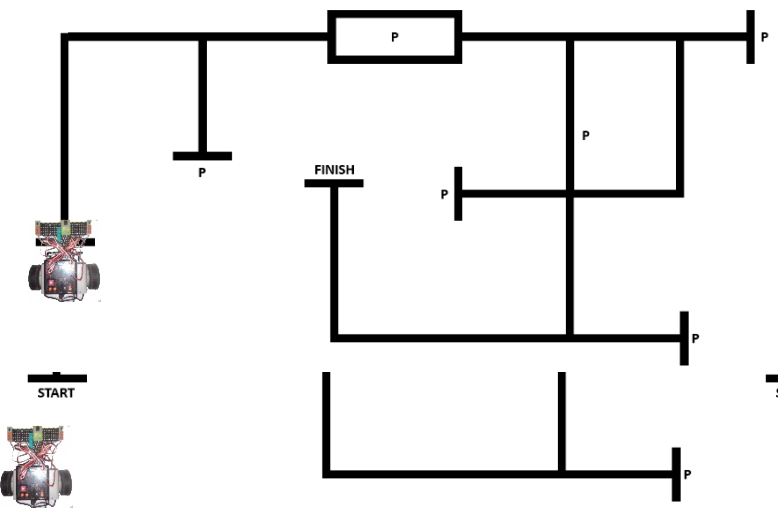


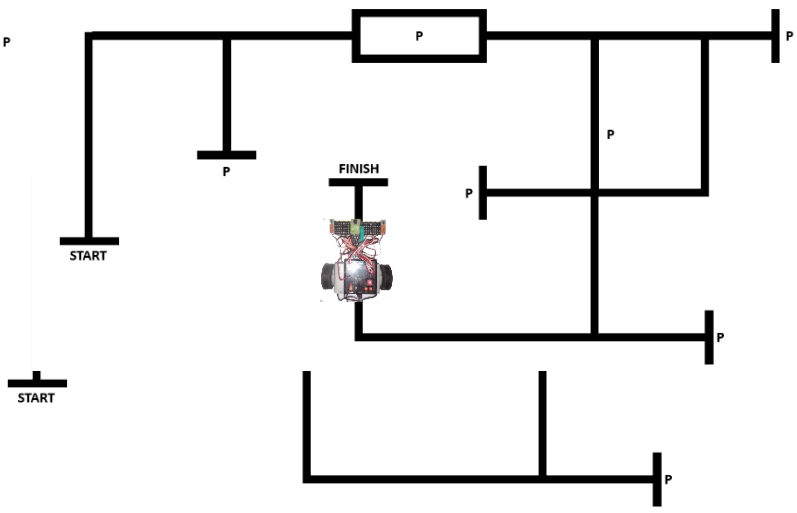


Sebelum program Setelah di program

3) Contoh program dari start => mengambil semua point => menuju garis Finish

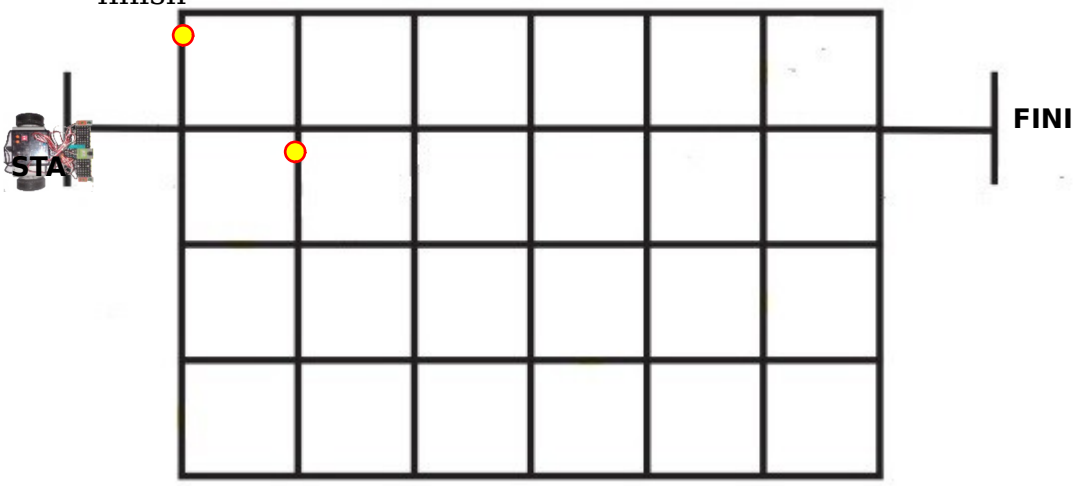
```
#include "functions_codes.h"
#include "maze.h"
int main ()
{
    mcu_initialization();
    ikutgarishitam(12,8,8);      //awal start
    kanan(10,5,250);
    ikutgarishitam(12,8,8);
    kanan(10,5,250);
    ikutgarishitam(24,8,8);      //dapat point ke 1
    balikkananhitam(8,-10);
    ikutgarishitam(24,8,8);
    kanan(10,5,250);
    ikutgarishitam(24,8,8);
    spartanhitam();              //dapat point ke 2
    ikutgarishitam(12,8,8);
    ikutgarishitam(12,8,8);
    ikutgarishitam(24,8,8);      //dapat point ke 3
    balikkananhitam(8,-10);
    ikutgarishitam(45,8,8);
    ikutgarishitam(45,8,8);
    kiri(5,10,250);
    ikutgarishitam(24,8,8);      //dapat point ke 4
    kanan(10,5,250);
    ikutgarishitam(24,8,8);      //dapat point ke 5
    balikkananhitam(8,-10);
    ikutgarishitam(24,8,8);
    kanan(10,5,250);
    ikutgarishitam(24,8,8);
    kiri(5,10,250);
    ikutgarishitam(24,8,8);      //dapat point ke 6
    balikkananhitam(8,-10);
    ikutgarishitam(12,8,8);
    ikutgarishitam(12,8,8);
    kanan(10,5,250);
    ikutgarishitam(24,8,8);      //sampai digaris finish
    stop(1000);
    return 0;
}
```





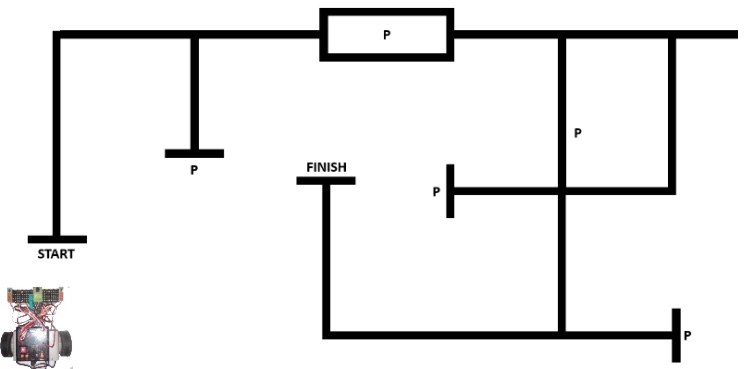
Sebelum program Setelah di program

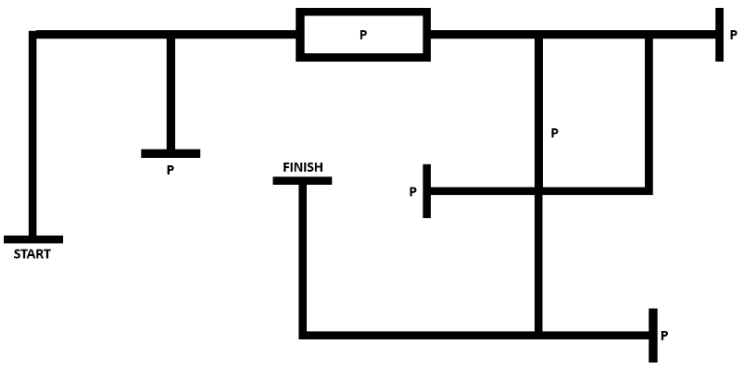
4) Mengambil semua point (lingkaran kining) tanpa menuju finish



Berikut programnya:

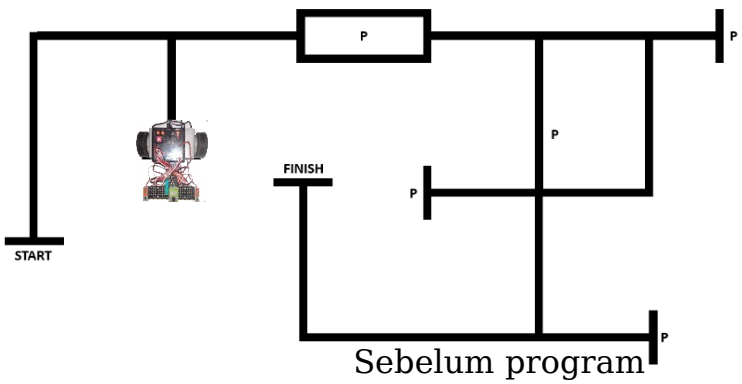
```
#include "functions_codes.h"
#include "maze.h"
int main ()
{
    mcu_initialization();
    ikutgarishitam(24,8,8);
    kiri(5,10,250);
    ikutgarishitam(12,8,8);
    kanan(10,5,250);
    ikutgarishitam(12,8,8);
    kanan(10,5,250);
    ikutgarishitam(24,8,8);
    stop(3000);
    return 0;
}
```



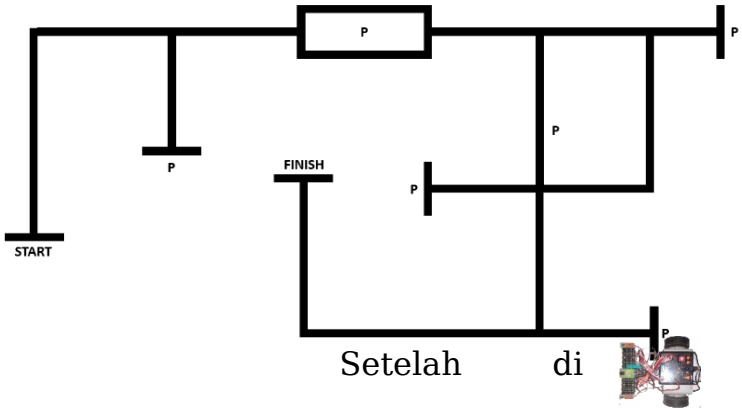


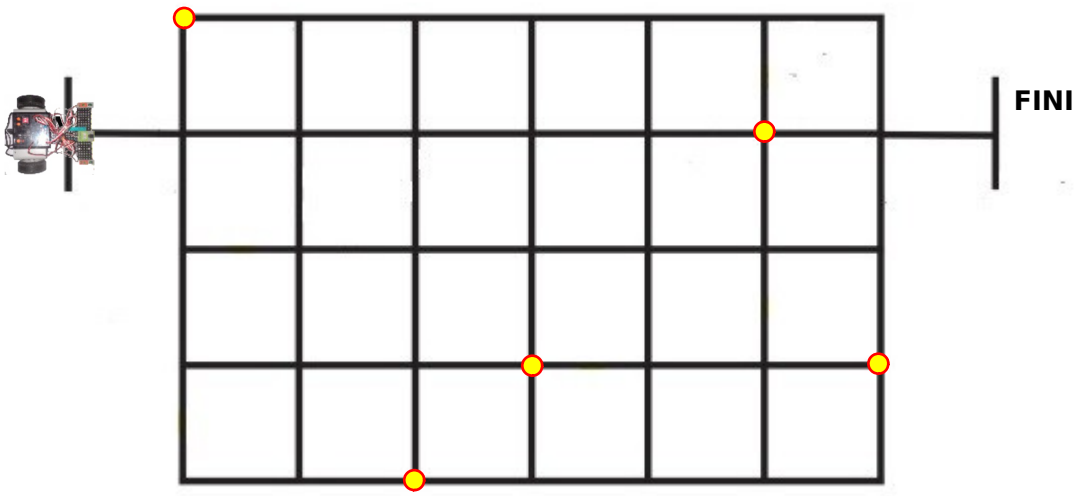
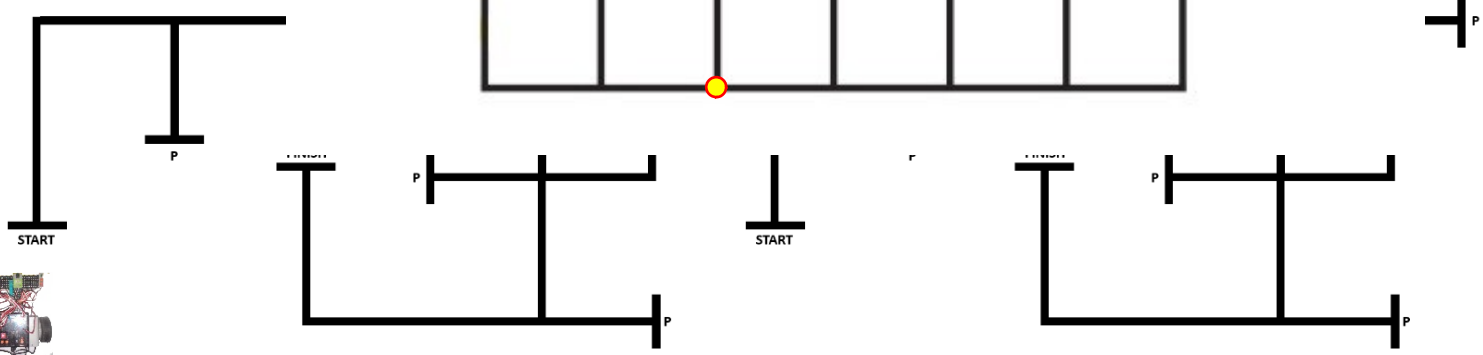
F. Latihan pemrograman

1. Buatlah program untuk menggerakan robot dari kondisi berikut:



Sebelum program



Setelah di program

2. Programlah robot kalian untuk mendapatkan semua point dan berhenti di garis Finish!